

WHAT IS CLAIMED IS:

1           1.     A method, comprising:  
2           monitoring paths between a first controller and second controller;  
3           determining whether one path has been unavailable for a predetermined time  
4 period in response to detecting that the path is unavailable;  
5           indicating the path in a first failed state if the path has been unavailable for more  
6 than the predetermined time period; and  
7           indicating the path in a second failed state if the path has not been unavailable for  
8 the predetermined time period.

1           2.     The method of claim 1, further comprising:  
2           indicating the path in a functioning state if the path is determined to be available.

1           3.     The method of claim 1, wherein the first failed state comprises a  
2 permanent failed state and the second failed state comprises a transient failed state.

1           4.     The method of claim 1, further comprising:  
2           receiving a write request;  
3           returning fail to the write request in response to determining that all paths are in  
4 the first failed state; and  
5           queuing the write request in a queue in response to determining that at least one  
6 path is in the second failed state and no paths are indicated in a functioning state.

1           5.     The method of claim 4, further comprising:  
2           submitting the write request to one path indicated in the functioning state to  
3 transmit to the secondary controller in response to determining that at least one path is in  
4 the functioning state.

1           6.     The method of claim 4, wherein at least one primary volume managed by  
2 the primary controller and at least one secondary volume managed by the secondary

3 controller are designated as volume pairs, wherein writes to one primary volume in one  
4 volume pair is copied to the corresponding secondary volume in the volume pair, further  
5 comprising:

6 suspending one volume pair including the primary volume to which the write  
7 request is directed in response to determining that all paths are in the first failed state.

1 7. The method of claim 4, further comprising:  
2 periodically processing the queue and write requests queued therein; and  
3 submitting the write requests in the queue to one path indicated in the functioning  
4 state to transmit to the secondary controller in response to determining that at least one  
5 path is in the functioning state

1 8. The method of claim 4, further comprising:  
2 indicating a time the write request was received when queuing the write request in  
3 the queue; and  
4 returning fail to one write request in the queue in response to determining that the  
5 write request has been queued longer than a request timeout period.

1 9. The method of claim 8, further comprising:  
2 periodically processing the queue and write requests queued therein to determine  
3 whether to return fail to those write requests queued longer than the request timeout  
4 period.

1 10. The method of claim 9, further comprising:  
2 determining whether at least one path is in a functioning state when periodically  
3 processing the queue, wherein fail is only returned to those write requests having been  
4 queued longer than the request timeout period in response to determining that no path is  
5 in the functioning state.

1           11.    The method of claim 1, further comprising:  
2           receiving a read request to access requested data;  
3           returning the requested data with the first controller in response to determining  
4   that the data is available at a first storage coupled to the first controller;  
5           determining that the requested data is not available at the first storage;  
6           returning fail to the read request in response to determining that all paths are in  
7   the first failed state in response to determining that the data is not available at the first  
8   storage; and  
9           queuing the read request in a queue to transfer to the secondary controller to  
10   access the requested data from a second storage in response to determining that at least  
11   one path is in the second failed state and no paths are indicated in a functioning state in  
12   response to determining that the data is not available at the first storage.

1           12.    The method of claim 1, further comprising:  
2           performing a failover to the second controller to service I/O requests through the  
3   second controller in response to detecting a failure related to the first controller;  
4           logging updates made by the second controller during the failover;  
5           transferring logged updates from the second controller to the primary controller in  
6   response to a failback to the first controller;  
7           returning fail to the transfer of one logged update to the first controller in response  
8   to determining that all paths are in the first failed state; and  
9           queuing one logged update to transfer to the first controller in a queue in response  
10   to determining that at least one path is in the second failed state and no paths are  
11   indicated in a functioning state.

1           13.    A system, comprising:  
2           a first controller;  
3           a second controller;  
4           paths between the first and second controller;  
5           code executed by the first controller to perform:

- 6 (i) monitoring paths between a first controller and second controller;  
7 (ii) determining whether one path has been unavailable for a  
8 predetermined time period in response to detecting that the path is unavailable;  
9 (iii) indicating the path in a first failed state if the path has been  
10 unavailable for more than the predetermined time period; and  
11 (iv) indicating the path in a second failed state if the path has not been  
12 unavailable for the predetermined time period.

1 14. The system of claim 13, wherein the code is executed by the first  
2 controller to further perform:  
3 indicating the path in a functioning state if the path is determined to be available.

1 15. The system of claim 13, wherein the first failed state comprises a  
2 permanent failed state and the second failed state comprises a transient failed state.

1 16. The system of claim 13, wherein the code is executed by the first  
2 controller to further perform:  
3 receiving a write request;  
4 returning fail to the write request in response to determining that all paths are in  
5 the first failed state; and  
6 queuing the write request in a queue in response to determining that at least one  
7 path is in the second failed state and no paths are indicated in a functioning state.

1 17. The system of claim 16, wherein the code is executed by the first  
2 controller to further perform:  
3 submitting the write request to one path indicated in the functioning state to  
4 transmit to the secondary controller in response to determining that at least one path is in  
5 the functioning state.

1           18.    The system of claim 16, wherein at least one primary volume managed by  
2   the primary controller and at least one secondary volume managed by the secondary  
3   controller are designated as volume pairs, wherein writes to one primary volume in one  
4   volume pair is copied to the corresponding secondary volume in the volume pair, and  
5   wherein the code is executed by the first controller to further perform:

6           suspending one volume pair including the primary volume to which the write  
7   request is directed in response to determining that all paths are in the first failed state.

1           19.    The system of claim 16, wherein the code is executed by the first  
2   controller to further perform:

3           periodically processing the queue and write requests queued therein; and  
4           submitting the write requests in the queue to one path indicated in the functioning  
5   state to transmit to the secondary controller in response to determining that at least one  
6   path is in the functioning state

1           20.    The system of claim 16, wherein the code is executed by the first  
2   controller to further perform:

3           indicating a time the write request was received when queuing the write request in  
4   the queue; and

5           returning fail to one write request in the queue in response to determining that the  
6   write request has been queued longer than a request timeout period.

1           21.    The system of claim 20, wherein the code is executed by the first  
2   controller to further perform:

3           periodically processing the queue and write requests queued therein to determine  
4   whether to return fail to those write requests queued longer than the request timeout  
5   period.

1           22.    The system of claim 21, wherein the code is executed by the first  
2   controller to further perform:

3           determining whether at least one path is in a functioning state when periodically  
4     processing the queue, wherein fail is only returned to those write requests having been  
5     queued longer than the request timeout period in response to determining that no path is  
6     in the functioning state.

1           23.     The system of claim 13, wherein the code is executed by the first  
2     controller to further perform:  
3           receiving a read request to access requested data;  
4           returning the requested data with the first controller in response to determining  
5     that the data is available at a first storage coupled to the first controller;  
6           determining that the requested data is not available at the first storage;  
7           returning fail to the read request in response to determining that all paths are in  
8     the first failed state in response to determining that the data is not available at the first  
9     storage; and  
10          queuing the read request in a queue to transfer to the secondary controller to  
11     access the requested data from a second storage in response to determining that at least  
12     one path is in the second failed state and no paths are indicated in a functioning state in  
13     response to determining that the data is not available at the first storage.

1           24.     The system of claim 13, further comprising:  
2     code executed by the second controller to perform:  
3           (i) performing a failover from the first controller to the second controller  
4     to service I/O requests through the second controller in response to detecting a  
5     failure related to the first controller;  
6           (ii) logging updates during the failover;  
7           (iii) transferring logged updates to the primary controller in response to a  
8     failback to the first controller;  
9           (iv) returning fail to the transfer of one logged update to the first controller  
10     in response to determining that all paths are in the first failed state; and

11 (v) queuing one logged update to transfer to the first controller in a queue  
12 in response to determining that at least one path is in the second failed state and  
13 no paths are indicated in a functioning state.

1 25. An article of manufacture for monitoring paths between a first controller  
2 and second controller, wherein the article of manufacture is capable of causing operations  
3 to be performed, the operations comprising:  
4 determining whether one path has been unavailable for a predetermined time  
5 period in response to detecting that the path is unavailable;  
6 indicating the path in a first failed state if the path has been unavailable for more  
7 than the predetermined time period; and  
8 indicating the path in a second failed state if the path has not been unavailable for  
9 the predetermined time period.

1 26. The article of manufacture of claim 25, wherein the operations further  
2 comprise:  
3 indicating the path in a functioning state if the path is determined to be available.

1 27. The article of manufacture of claim 25, wherein the first failed state  
2 comprises a permanent failed state and the second failed state comprises a transient failed  
3 state.

1 28. The article of manufacture of claim 25, wherein the operations further  
2 comprise:  
3 receiving a write request;  
4 returning fail to the write request in response to determining that all paths are in  
5 the first failed state; and  
6 queuing the write request in a queue in response to determining that at least one  
7 path is in the second failed state and no paths are indicated in a functioning state.

1           29.    The article of manufacture of claim 28, wherein the operations further  
2   comprise:  
3           submitting the write request to one path indicated in the functioning state to  
4   transmit to the secondary controller in response to determining that at least one path is in  
5   the functioning state.

1           30.    The article of manufacture of claim 28, wherein at least one primary  
2   volume managed by the primary controller and at least one secondary volume managed  
3   by the secondary controller are designated as volume pairs, wherein writes to one primary  
4   volume in one volume pair is copied to the corresponding secondary volume in the  
5   volume pair, wherein the operations further comprise:  
6           suspending one volume pair including the primary volume to which the write  
7   request is directed in response to determining that all paths are in the first failed state.

1           31.    The article of manufacture of claim 28, wherein the operations further  
2   comprise:  
3           periodically processing the queue and write requests queued therein; and  
4           submitting the write requests in the queue to one path indicated in the functioning  
5   state to transmit to the secondary controller in response to determining that at least one  
6   path is in the functioning state

1           32.    The article of manufacture of claim 28, wherein the operations further  
2   comprise:  
3           indicating a time the write request was received when queuing the write request in  
4   the queue; and  
5           returning fail to one write request in the queue in response to determining that the  
6   write request has been queued longer than a request timeout period.

1           33.    The article of manufacture of claim 32, wherein the operations further  
2   comprise:



3           periodically processing the queue and write requests queued therein to determine  
4   whether to return fail to those write requests queued longer than the request timeout  
5   period.

1           34.    The article of manufacture of claim 33, wherein the operations further  
2   comprise:  
3           determining whether at least one path is in a functioning state when periodically  
4   processing the queue, wherein fail is only returned to those write requests having been  
5   queued longer than the request timeout period in response to determining that no path is  
6   in the functioning state.

1           35.    The article of manufacture of claim 25, wherein the operations further  
2   comprise:  
3           receiving a read request to access requested data;  
4           returning the requested data with the first controller in response to determining  
5   that the data is available at a first storage coupled to the first controller;  
6           determining that the requested data is not available at the first storage;  
7           returning fail to the read request in response to determining that all paths are in  
8   the first failed state in response to determining that the data is not available at the first  
9   storage; and  
10          queuing the read request in a queue to transfer to the secondary controller to  
11   access the requested data from a second storage in response to determining that at least  
12   one path is in the second failed state and no paths are indicated in a functioning state in  
13   response to determining that the data is not available at the first storage.

1           36.    The article of manufacture of claim 25, wherein the operations further  
2   comprise:  
3           performing a failover to the second controller to service I/O requests through the  
4   second controller in response to detecting a failure related to the first controller;

5 logging updates made by the second controller during the failover;  
6 transferring logged updates from the second controller to the primary controller in  
7 response to a failback to the first controller;  
8 returning fail to the transfer of one logged update to the first controller in response  
9 to determining that all paths are in the first failed state; and  
10 queuing one logged update to transfer to the first controller in a queue in response  
11 to determining that at least one path is in the second failed state and no paths are  
12 indicated in a functioning state.